Embedded Control of Electric Vehicle Motor System

EGH456 Embedded Systems

Problem Based Learning Assignment

Assignment Report due: 11:59pm Friday Week 13

Assignment Demonstration due: Week 13 at a scheduled time for your group

Weight: 30%

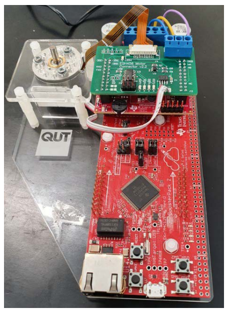
The specifications in this document are subject to minor changes if further clarification is needed

1. Problem Description

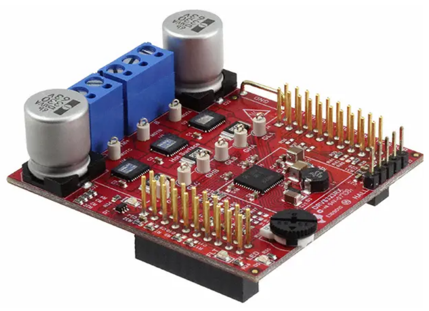
Battery-Electric and Hybrid-Electric vehicles are becoming more and more economical, efficient and environmentally friendly resulting in an increase in adoption rates. In addition, the recent advancements in autonomous electric vehicles are also resulting in the automotive industry starting to transition from petrol to electric vehicle design. As an embedded system engineer you have been contracted by an automotive manufacture who is transitioning from petrol to Battery-Electric technology. Your task is to design the real time sensing, motor control and user interface for the electric vehicle.

Your goal is to develop the embedded software to safely control and monitor an electric vehicle including the sensing and actuation of a 3-phase Brushless DC (BLDC) motor commonly used in electric vehicles. This will include monitoring the state of the motor such as velocity, power and temperature and handle the start-up, braking and emergency procedures for the system. Your system design will require real-time handling of multiple tasks for sensor acquisition and filtering, motor fault handling and system display of critical information. Your setup includes a motor testing station including the motor driver and sensor board and a development kit for the Tiva TM4C1294NCPDT microcontroller. Figure 1b illustrates the setup interface of the testing station. Inputs and outputs of sensors and per-phase connections are compatible with the requirements of the microcontroller ports, with an electrical interface to achieve this.

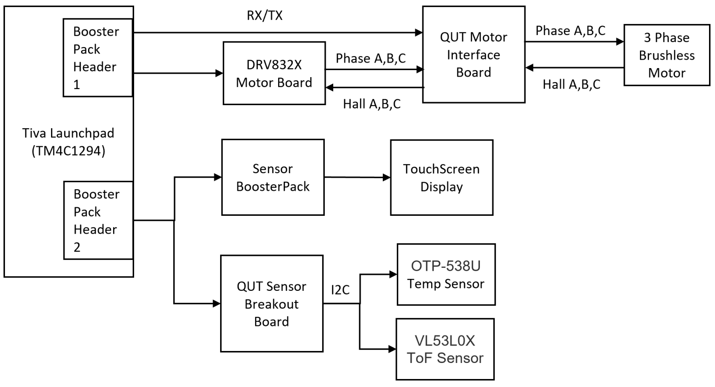
The mapping of the microcontroller pins (GPIO, I2C, ADC, UART) and the motor driver inputs and outputs is provided in a separate document. Additional resources that will be provided for this assignment includes a description of the operation of the motor, a custom motor driver library (MotorLib) and a motor kit setup guide.



(a)



(b)



(c)

Figure 1: (a) Simulated Electric Vehicle Sensor and Motor Testing Station. The system includes a microcontroller development kit, a motor driver and sensor board, brushless motor with hall effect sensors for position and velocity feedback and temperature and current sensors for fault monitoring. (b) BOOSTXL-DRV8323RH Motor Control Launchpad Boosterpack. (c) a simplified block diagram of the hardware used within this assignment. Please see the additional assignment reference document for a more detailed pinout of the Boosterpack headers

2. Requirements

The assignment can be separated into three main sections: Motor Control, Sensing and User Interface. Each section has different requirements and are described below. You are required to design the solution using a software driver model for each subsystem. The final software will then use the driver functionalities to integrate each subsystem into one coherent solution for demonstration. Your team should coordinate and design the software system together in order to manage events, shared resources and CPU utilisation, taking into account the priority of each subsystem and their tasks.

2.1 Motor Control

You are to write a device driver for the DRV832x motor driver board (including the additional custom motor adapter board) to provide an Application Programming Interface (API) that will:

1. Start and control the motor safely, handling any fault conditions using the provided state transition diagram (see supporting documents for how to control the phases of the motor)
2. Control a user given desired speed of the motor in revs/min
3. Safely start and accelerate the motor to the desired speed (See below for acceleration specifications)
4. Safely decelerate the motor (See below for deceleration specifications)

The motor must be controlled to match a desired speed/RPM modified by the user interface (this emulates the driver of the vehicle pressing the foot pedal). This requires measuring the speed of the motor and changing the PWM duty cycle to match the desired speed. The actual motor speed can change depending on the load on the motor and is not constant for a constant PWM Duty value. Less marks will be awarded if the motor speed is not controlled by measuring the motors current speed (aka as an open loop controller). In order to safely start and brake the vehicle (motor) the system should control the speed of the motor following acceleration and deceleration limits (See separate reference document for supporting speed control diagram). You will also need to investigate how to effectively get a 3 phase brushless motor to start correctly. A third-party motor driver library will also be provided to help students setup the GPIO pins and correctly drive the phases of the brushless motor. The system will also require different deceleration limits depending on whether an emergency condition has occurred or a user has input a slower desired speed. The acceleration and deceleration limits are defined as:

1. Motor Acceleration $\leq 500 RPM/s$
2. Motor Deceleration (setting slower speed) $\leq 500 RPM/s$
3. Motor Deceleration (emergency stop aka e-stop) $= 1000 RPM/s$

Please ensure you handle units correctly as acceleration and deceleration values will be tested. Note, the deceleration for e-stop should be equal to (not greater or less than) the desired value as we would like the vehicle to come to a complete stop quickly but safely.

2.1.1 E-stop Conditions

The following conditions should cause the motor control to enter into an e-stop condition causing the motor to safely brake using the previously indicated deceleration limit. For the temperature, acceleration and heading conditions, only implement the condition if you have selected the sensor in section 2.3.

1. Total motor current has exceeded a user defined threshold
2. Temperature has exceeded a user defined threshold
3. Absolute acceleration has exceeded a user defined threshold (an impact has occurred) measured via the accelerometer sensor (see section 2.3)
4. Heading of vehicle has exceeded a user defined upper and lower threshold (the car is sideways) measured via the compass sensor (see section 2.3)
5. Distance has gone below a user defined threshold (vehicle is about to hit an obstacle) measured via the Time of Flight distance sensor (see section 2.3)

2.2 Motor Library and Motor Kits

A custom motor library will be supplied to your group in order for you to easily and safely commutate (switch) the phases of the motor. This will be available via Blackboard and separate instructions will be supplied describing how to install and use the library. You must use this motor library to minimise the risk of damaging the motor kits supplied with this assignment. However, this does not guarantee that the motors will not be damaged and it is recommended that the motor kits are monitored during operation and if debugging your software while operating the motors that either the motor kits are powered down or monitored such that the do not heat up significantly. The teaching team have added some protection mechanisms (motor adapter board and current limited power supplies) to minimise this risk but with all electrical and mechanical hardware it is almost impossible to protect against all possible failures.

2.3 Sensing

In order to monitor critical information on the state of the vehicle you will need to write a device driver for the various sensors available on the vehicle. You will need to write the sensor drivers in order to provide an Application Programming Interface (API) to perform the following functionalities:

1. Power Sensing - Read and filter two motor phase currents via the analogue signals provided by the current sensors on the DRV8323 board (only two sense lines are available as ADC inputs to the microcontroller). Using the motor current and motor voltage (24V nomimal) calculate the power usage of the motor.
2. Speed Sensing - Measure and filter the current speed of the motor in revs/min (rpm)

Choose 2 out of the 5 following sensors to add to your project:

1. Vechicle acceleration (BMI160) - Read and Filter the acceleration on all three axis and calculate average absolute acceleration (sum of absolute acceleration in each axis). This can be used to detect a sudden crash.
2. Temperature (OTP-538U) - Read and filter the temperature from the temperature sensor over via a I2C connection provided by the QUT sensor breakout board.

3. Compass (BMM150) - Read and filter the magnetometer on all three axis and calculate the heading of the vehicle from the x and y axis.
4. Light Level - Read and filter the light level from the OPT3001 ambient light sensor over an I2C connection.
5. Distance (VL53L0X) - Read and filter the Time of Flight (ToF) distance sensor over an I2C connection.

2.3.1 Data Filtering

Sensor data usually suffer from measurement noise. In order to correctly identify faults in a real-time fashion and ensure noise does not trigger false alarms you must filter this data before displaying it. You must filter the data using the following specifications:

- Current (DRV8323) – sample buffer size of 5 or more, sampled at a rate $\geq$ 150 Hz
- Speed (Hall Sensors) – sample buffer size of 5 or more sampled at a rate $\geq$ 100 Hz
- Accelerometer (BMI160) - sample buffer size (of each axis) of 5 or more sampled at a rate $\geq$ 200 Hz
- Temperature (OTP-538U) – sample buffer size of 3 or more sampled at a rate $\geq$ 1 Hz.
- Compass (BMM150) - sample buffer size (of each axis) of 3 or more sampled at a rate $\geq$ 10 Hz
- Light (OPT3001) – sample buffer size greater than 5 sampled at a rate $\geq$ 2 Hz
- Distance (VL53L0X) – sample buffer size greater than 5 sampled at a rate $\geq$ 5 Hz

It is recommended that this filtering should not be done in a hardware interrupt. For example for current sensing, you should capture 1 sample of the current every 6.67ms (150Hz) and keep a buffer (window size) of 5 samples. You may use your own filtering technique such as a sliding window technique (see sliding window for examples). It is recommended that once a set of samples have filled the buffer then a software interrupt should be used to perform the follow up filtering update step. If you believe more samples at higher frequency would improve the performance of the filter then it is okay to increase these specifications.

2.4 User Interface

You are required to develop a graphical user interface to control the operation of the electric vehicle and display the state of the vehicle. This includes displaying critical information and including the ability to change the status of the vehicle. You should consider the layout of the interface for easy control and display of the information.

A list of specific features of the user interface is defined below:

1. Starting and stopping the motor using buttons (on-screen or push button)
2. Displaying start and stopped status using an LED
3. Setting the desired speed of the motor using user input
4. Keeping track of time and displaying a calendar time (date can be incorrect but time needs to update)

A feature of the user interface is the ability to set upper and lower limits which then throw an e-stop condition if outside these limits. The following limits should be implemented and have the ability to be modified via the user interface.

1. Setting an upper limit of allowable current drawn by the motor. If this condition is met a fault has occurred, an e-stop condition should be flagged and the motor should be stopped safely

Implement the following sensor specific features depending on which 2 sensors you selected out of the 4 given from section 2.3:

1. Light - Classify and display whether it is currently day or night (night time is classified as the ambient light being less than 5 lux). Turn on an LED when the time of day is classified as night time (simulating auto headlights).
2. Temperature - Setting an upper limit of allowable temperature of the motor. If this condition is met a fault has occurred and the motor should be stopped safely
3. Acceleration - Set up an upper limit of allowable acceleration of the vehicle. If this condition is met a fault has occurred, an e-stop condition should be flagged and the motor should be stopped safely.

4. Compass - Set up an upper and lower limit of allowable heading of the vehicle. If this condition is met a fault has occurred, an e-stop condition should be flagged and the motor should be stopped safely.
5. Distance - Set up a lower limit of allowable distance to an obstacle in front of the vehicle. If this condition is met a crash is about to occur and an e-stop condition should be flagged and the motor should be stopped safely.

On a second page/tab within the user interface you will be required to plot the filtered sensor information from the vehicle in a real-time, user friendly manner. Use your own design/structure for displaying the sensor information, but it must display the following information as a moving plot over time:

1. motor speed (rpm)
2. approximate power usage of the motor using the phase current measurements (in watts)

Only display the following 2 sensors that were chosen to be implemented from section 2.3

1. Light - ambient light level (in Lux)
2. Temperature - current temperature (in Degrees Celsius)
3. Acceleration - Average absolute acceleration from the accelerometer (m/s^2)
4. Compass - Heading of the vehicle (in degrees)
5. Distance - Distance to any object in front of the vehicle (in metres)

For displaying the sensor data, units can be of different magnitudes (e.g. Kilowatts), scaled or adjusted for display purposes.

3. Demonstration

You are required to present and demonstrate your work to the teaching team. The groups will present as a team; however, you will be individually assessed. For the demonstration, you should prepare to show and talk about what you have done. You will also be asked questions throughout your demonstration. Additional advanced features will also improve your mark only if the design requirements for that system have been met. For the presentation you should aim to present your designed system to the client which is the automotive manufacturing company (teaching team). Your presentation should include an overall description of your final design, how you have met specifications and showing all relevant details.

The duration is strictly 3 minutes per student for presentation (come prepared for this) and 13 minutes for demonstration. You should prepare only one PowerPoint slide per student.

4. Assessment

4.1 Mark Distribution

Presentation (5%) - Based on effectiveness of communication, use of visuals and quality of the presentation by each student.

Achievement (15%) - Based on demonstration of evidence that requirements are met.

Report (10%) - A suggested report format is provided at the end of this document.

Criterion Referenced Assessment Sheet – CRA sheets for the demonstration and report will be available on Blackboard

4.2 Late Demonstration

This will not be permitted except for cases permitted by QUT policy. Regardless of the state of completion of your assignment work, you are required to make a presentation as scheduled describing your work up to that time.

4.3 Penalty for Late Submission

QUT policy for late submission of assignments will apply. A late demonstration and/or report without an approved exemption will receive 0%.

4.4 Requirements that could not be demonstrated

If you describe your work well enough – and your demonstration marks were low – you could be given partial credit to compensate for an inability to demonstrate when it was required.

You can add an explanation for each failure if you have been able to figure out how you might proceed if you could do it again.

5. Suggested Format of the Report

The report should contain the following items. The content and style are flexible if these are separately identifiable. Approximate page guidelines are given in parentheses.

- ***Cover page*** – names, student ID numbers, course, and unit information. (1 page)
- ***Table of Contents*** – Section headings, list of figures, list of tables. (1 to 2 pages)
- ***Introduction*** – Problem statement, context, requirements, and statements on the individual contributions by each team member. (2 pages)
- ***Design and Implementation*** – Approach to design, important issues and choices and their relationships to theoretical concepts and the hardware and software platforms. Check if you have addressed:
  - Provided a graphical representation for your design?
  - Provided a Project Folders and Files diagram?
  - Did you use the GPIO module? How? Did you use the graphics library? How? Did you use Tasks, Hwi and Swi? How?
  - Did you use multiple threads/handlers? How? Did you use the ADC? How?
- ***Results*** – Summary of evidence of functional requirements that were demonstrated and explanation of failures as learning outcomes in terms of what could have been done differently. (2 pages)
- ***References*** – Including vendor supplied technical documents with a table that links each document to sub-sections of your report where they are relevant with entries: The reference number, the sub-section number, topic keyword or keywords, the pages that were found useful. (2 pages)
- ***Appendix*** – Mention the name of the hardware development platform and versions of the tools used such as Code Composer Studio, Tivaware, TI-RTOS, etc. (1 page)

Reference Documents

- DRV832x 6 to 60-V Three-Phase Smart Gate Driver Datasheet
- DRV832XX EVM Sensored Software User's Guide
- BOOSTXL-DRV8323Rx EVM User's Guide
- TI-RTOS User's Guide
- TI-RTOS SYS/BIOS User's Guide
- EK-TM4C1294XL User's Guide
- TM4C1294NCPDT Datasheet
- OPT3001 Ambient Light Sensor Datasheet
- BOOSTXL-SENSORS Sensors BoosterPack User's Guide
- Grove IR Temperature Sensor - OTP-538U
- Grove Time of Flight Distance Sensor - VL53LOX